МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ   
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Информационных систем и технологий

Специальность 1-40 05 01 «Информационные системы и технологии»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА КУРСОВОГО ПРОЕКТА**

по дисциплине «Компьютерные языки разметки»

Тема Web-сайт: «Интернет-магазин бытовой техники»

**Исполнитель**

студент(ка) 1 курса 2 группы \_\_\_\_\_\_\_\_\_\_\_\_ В.О. Лемешевский

подпись, дата

**Руководитель**

преподаватель-стажёр \_\_\_\_\_\_\_\_\_\_\_ Л.П. Степанова

должность, ученая степень, ученое звание подпись, дата

Допущен(а) к защите \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата, подпись

Курсовой проект защищен с оценкой

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Л.П. Степанова

подпись дата инициалы и фамилия

Содержание

[Введение 4](#_Toc134583406)

[1. Постановка задачи 5](#_Toc134583407)

[1.1 Обзор аналогичных решений 5](#_Toc134583408)

[1.2 Техническое задание 9](#_Toc134583409)

[1.3 Выбор средств реализации программного продукта 9](#_Toc134583410)

[1.4 Выводы 10](#_Toc134583411)

[2. Проектирование страниц веб-сайта 10](#_Toc134583412)

[2.1 Выбор способа вёрстки 10](#_Toc134583413)

[2.2 Выбор стилевого оформления 10](#_Toc134583414)

[2.3 Выбор шрифтового оформления 11](#_Toc134583415)

[2.4 Разработка логотипа 11](#_Toc134583416)

[2.5 Разработка пользовательских элементов 12](#_Toc134583417)

[2.6 Разработка спецэффектов 12](#_Toc134583418)

[2.7 Выводы 13](#_Toc134583419)

[3. Реализация структуры веб-сайта 14](#_Toc134583420)

[3.1. Структура HTML-документа 14](#_Toc134583421)

[3.2. Добавление таблиц стилей SCSS и CSS 18](#_Toc134583422)

[3.3. Использование стандартов XML (SVG) 22](#_Toc134583423)

[3.4 Выводы 23](#_Toc134583424)

[4. Тестирование веб-сайта 24](#_Toc134583425)

[4.1. Адаптивный дизайн веб-сайта 24](#_Toc134583426)

[4.2. Кроссбраузерность веб-сайта 26](#_Toc134583427)

[4.3. Руководство пользователя 27](#_Toc134583428)

[4.4. Валидность 29](#_Toc134583429)

[4.4 Выводы 29](#_Toc134583430)

[Заключение 30](#_Toc134583431)

[Список использованных литературных источников 31](#_Toc134583432)

[Приложение А Прототипы веб-страниц 32](#_Toc134583433)

[Приложение Б Макет структуры веб-сайта 34](#_Toc134583434)

[Приложение В Листинг HTML-документа 35](#_Toc134583435)

[Приложение Г Листинг SCSS и CSS 41](#_Toc134583436)

[Приложение Д Листинг кода для работы с XML-файлами 45](#_Toc134583437)

[Приложение Ж Листинг JavaScript 50](#_Toc134583438)

# Введение

Интернет-магазины бытовой техники имеют высокую актуальность в настоящее время благодаря нескольким факторам.

Во-первых, они предоставляют потребителям удобный способ покупки товаров, не выходя из дома, что особенно важно в условиях пандемии.

Во-вторых, в интернет-магазинах широкий ассортимент товаров, их наличие и цены могут быть легко сравнены с другими магазинами.

В-третьих, многие интернет-магазины предлагают дополнительные услуги, такие как бесплатная доставка, установка и обслуживание, что делает процесс покупки еще более простым и удобным для покупателей.

Кроме того, развитие технологий и использование аналитических инструментов помогает интернет-магазинам предлагать персонализированные предложения и удовлетворять потребности разных категорий клиентов.

В целом, интернет-магазины бытовой техники являются важным элементом рынка и продолжают расти в популярности среди покупателей.

Цель курсовой работы:разработать веб-сайт интернет-магазина с использованием HTML5 и XML, а также с применением CSS3 для стилей и JavaScript для динамики.

Задачи:

1. Проанализировать существующие языки разметки, инструменты и библиотеки для создания веб-сайта.
2. Разработать макет и прототип сайта.
3. Разработать структуру веб-сайта.
4. Наполнить сайт информацией по теме.
5. Протестировать веб-сайт.
6. Разработать руководство пользователя.

# 1. Постановка задачи

## 1.1 Обзор аналогичных решений

Перед тем как приступать к разработке своего веб-сайта можно изучить уже готовые решения и посмотреть, что по мнению других разработчиков является наиболее необходимым для пользователей, чтобы приложение действительно было полезным и удобным.

В качестве аналогов были выбраны сайты интернет-магазинов и онлайн каталогов, а именно: 5element.by [1], sila.by [2], 21vek.by [3]. Данные сайты имеют необходимый функционал.

В качестве первого аналога рассмотрим интернет-магазин “5 элемент”.

На рисунке 1.1 представлено оформление сайта.

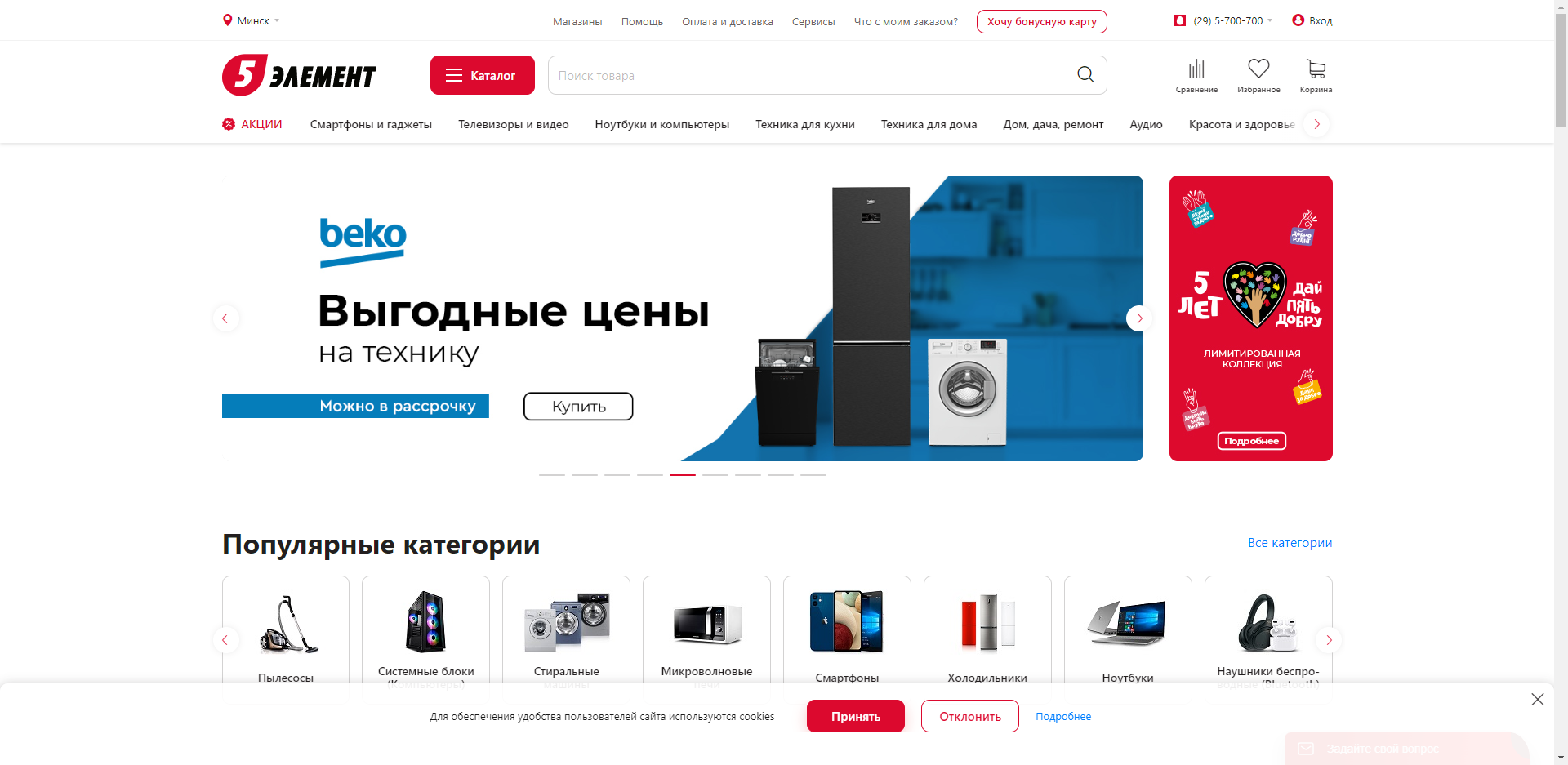


Рисунок 1.1 – Оформление сайта “5 элемент”.

Это интернет-магазин, специализирующийся на продаже товаров для здоровья и красоты. На сайте представлен широкий ассортимент товаров: от продуктов питания и биологически активных добавок до косметики и средств для ухода за волосами.

Кроме того, на сайте можно купить товары для дома и спорта, а также товары для детей и животных. Сайт работает по всей Беларуси и предлагает доставку товаров по всей стране. Клиенты могут оплачивать заказы онлайн или при получении наличными или банковской картой.

Сайт построен на базе CMS OpenCart и использует HTML, CSS и JavaScript для реализации пользовательского интерфейса. Дизайн сайта выполнен в современном минималистичном стиле с использованием ярких цветов, которые позволяют выделить важные элементы интерфейса.

На главной странице сайта представлены разделы с категориями товаров, специальными предложениями, новинками и бестселлерами. В верхней части сайта расположены ссылки на основные разделы сайта, а также корзина и форма поиска.

Каждая страница категории товаров содержит фотографии продуктов, их описание, цену и возможность добавить их в корзину. Кроме того, на сайте представлены страницы с информацией о доставке, оплате, контактах и часто задаваемых вопросах. Сайт адаптивный и имеет удобный мобильный

интерфейс. Дизайн представлен на рисунке 1.2.

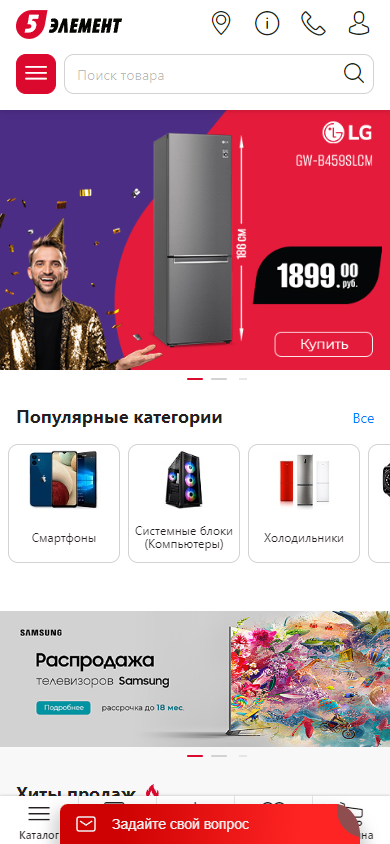


Рисунок 1.2 – Мобильная версия сайта интернет-магазина “5 элемент”

Следующий сайт – интернет-магазин sila.by. Оформление представлено на рисунке 1.3.

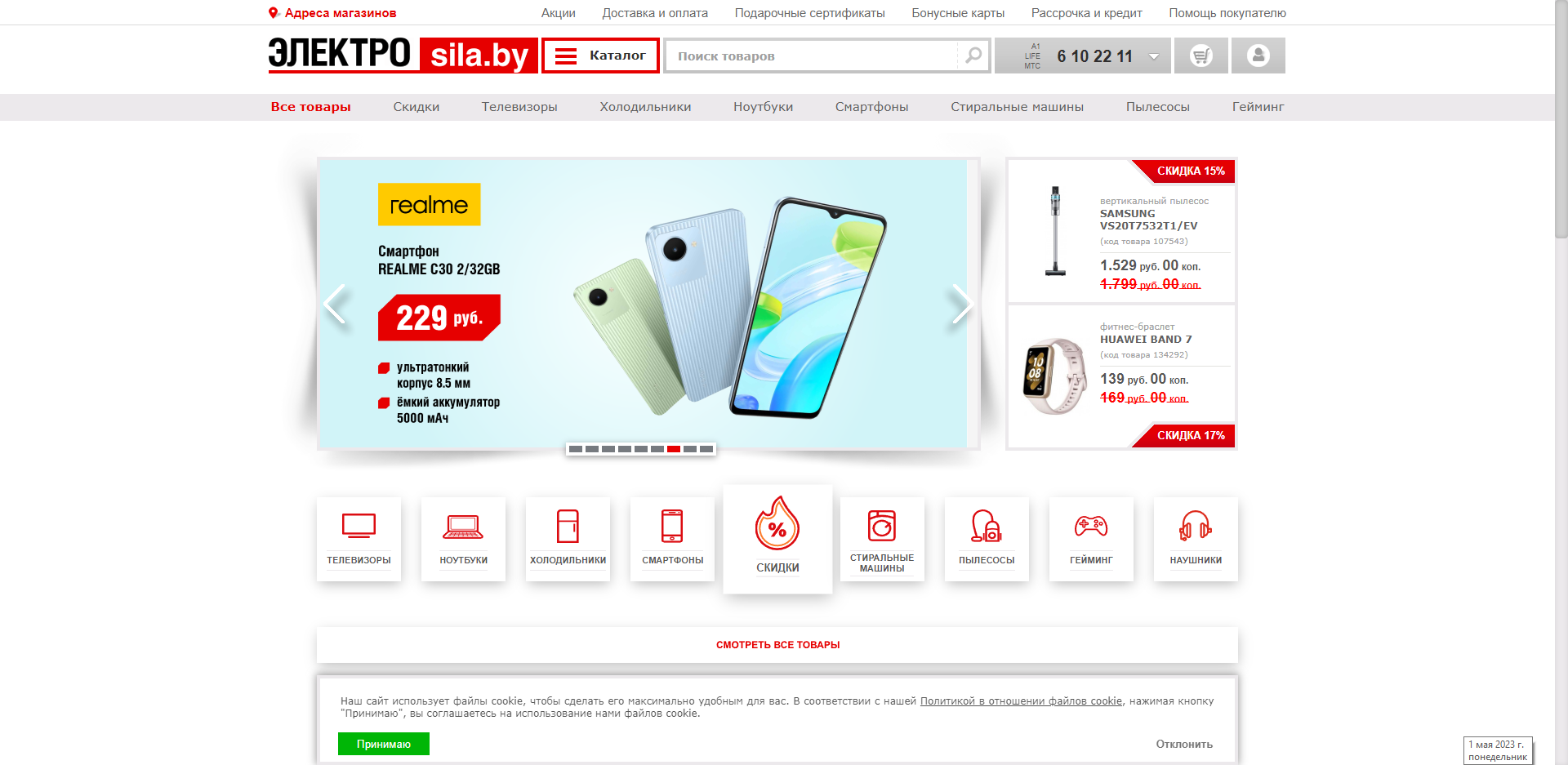


Рисунок 1.3 – Окно главной страницы сайта sila.by

Сайт https://sila.by/ является интернет-магазином, который предлагает широкий ассортимент товаров для дома и сада, а также инструменты и оборудование для строительства и ремонта.

На сайте можно найти товары таких категорий, как электроинструменты, сварочное оборудование, насосы, садово-огородная техника, сантехника, отопление и многое другое.

Также на сайте предоставляются услуги доставки и установки товаров. Сайт выглядит удобным и понятным в использовании, с информативными описаниями товаров и возможностью сравнения их характеристик.

Сайт https://sila.by/ имеет адаптивную верстку, что позволяет его корректно отображать на устройствах разных размеров и разрешений экрана.

Верстка сайта выполнена в светлых тонах, с использованием современных технологий и стилей. Шапка сайта содержит логотип, основное меню, поиск и контактную информацию. На главной странице расположен большой слайдер с актуальными предложениями, а также блоки с различными категориями товаров. Дизайн данного сайта мне нравится меньше из-за многоструктурированности.

Дизайн страниц товаров достаточно прост и удобен в использовании, с информативными описаниями и изображениями товаров, а также возможностью выбрать определенные параметры и добавить товар в корзину.

Аналог №3 - 21vek.by

Последним был выбран сайт 21vek.by, фотография его страницы представлена на рисунке 1.5.

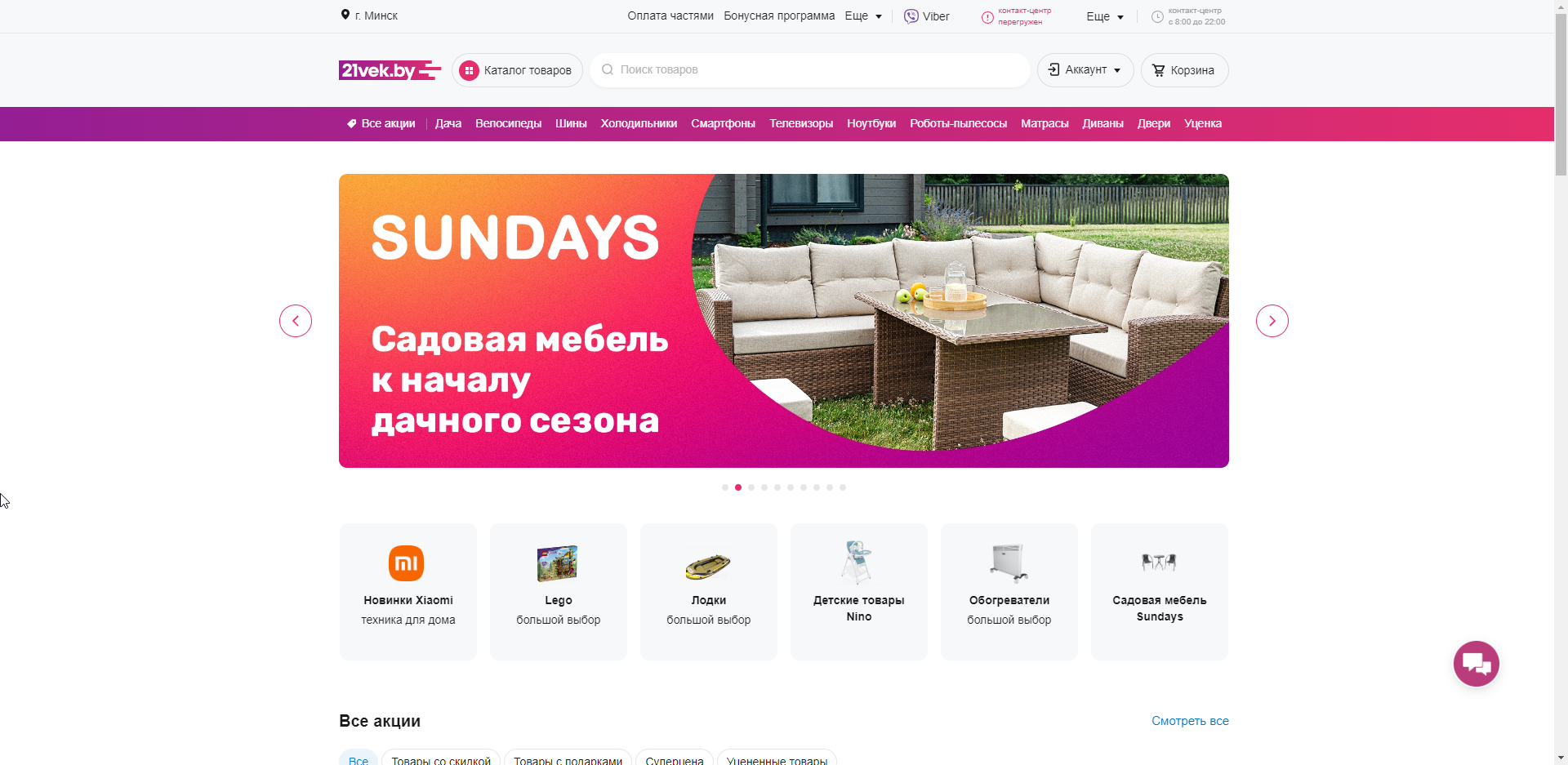


Рисунок 1.4 - Оформление сайта «21vek.by»

У сайта простая и приятная цветовая гамма, используются только 4 цвета, карточки товаров выглядят лаконично, вся информация о товаре, как таковые, анимации отсутствуют, что делает использование сайта, имеется мобильная версия, в которой реализована адаптивность, в компьютерной версии она отсутствует, однако вариант версии определяется только по useragent-у браузера, из-за чего могут быть проблемы при использовании.

При анализе функционала были выявлены следующие преимущества:

* Единые цвета.
* Понятный интерфейс.
* Интерактивность.

К недостаткам в функционале веб-сайта можно отнести:

* Отсутствует адаптивность верстки у компьютерной версии, версия выбирается исключительно по useragent-у.
* малый размер шрифта в разделе с информацией о товаре.

## 1.2 Техническое задание

Поставлена задача реализовать полноценный многостраничный веб-сайт на тему “Интернет-магазин бытовой техники”. На сайте должна быть представлена информация о бытовых товарах, такая как текущие скидки, категории, каталог товаров, возможность добавлять товар в корзину, возможность прочитать информацию о сайте и обратиться за помощью.

Для привлечения внимания аудитории главная задача сделать удобный интерактивный сайт с подробной информацией о продукте и актуальной информацией.

Основными задачами сайта являются:

1. Предоставление навигационного меню.
2. Предоставление информации о товаре.
3. Предоставление информации о цене.

На главной странице меню будет расположено сверху. В меню будут предоставляться ссылки на основную информацию.

Веб-страница должна быть кроссбраузерной и адаптивной. Начиная с ширины в 900px должна открываться мобильная версия.

## 1.3 Выбор средств реализации программного продукта

Курсовая работа выполняется в редакторе кода Visual Studio Code.

При создании веб-страниц были использованы языки: HTML, CSS, JS, XML.

HTML (HyperText Markup Language) — стандартный язык разметки гипертекстовых страниц в Интернете. Страницы успешно интерпретируются браузерами, которые отображают их на экранах различных электронных устройств в удобном для человека виде.

CSS — это фактически язык стилей, который определяет отображение HTML-документов. CSS работает со шрифтами, с цветами символов и фона, с полями, со строками, с высотой и с шириной элементов отображения, с фоновыми изображениями, с позиционированием элементов и со многим другим.

Если HTML необходим для структурирования содержания страницы, то CSS необходим для того, чтобы форматировать это структурированное содержание.

JavaScript — это язык, который позволяет применять сложные вещи на веб-странице — каждый раз, когда на ней происходит что-то большее, чем просто её статичное отображение — без JavaScript не обошлось. Во все основные браузеры встроен интерпретатор JavaScript, именно поэтому они могут выполнять скрипты на странице. JavaScript можно использовать не только в браузере, это полноценный язык, программы на котором можно запускать и на сервере.

## 1.4 Выводы

В данном разделе были рассмотрены аналоги на выбранную тему сайта для предотвращения ошибок в реализации веб-сайта. Были определены основные технические задания по созданию страницы. Также выбраны средства реализации программного продукта, а именно языки HTML, CSS, JS.

# 2. Проектирование страниц веб-сайта

## 2.1 Выбор способа вёрстки

Требования к проекту включают адаптивность и кроссбраузерность, поэтому была мной выбрана flex-верстка, которая будет применятся для основных элементов страницы.

Flex – это новая технология, которая уже имеет достаточно широкую поддержку браузеров. Flexbox предоставляет инструменты для быстрого создания сложных, гибких макетов, и функции, которые были сложны в традиционных методах CSS.

## 2.2 Выбор стилевого оформления

Для реализации стилевого оформления сайта было выбрано лаконичное оформление в сдержанных цветах с элементами классического веб-дизайна. Такой стиль позволяет привлечь внимание пользователей отсутствием лишних деталей. На сайте будет представлена лишь необходимая информация. Основной акцент сделан на контекст сайта.

Цветовые решения не слишком яркие. Основные цвета: #000000, #d8d8d8 и #97d3ff. Цветовая палитра представлена на рисунке 2.1



Рисунок 2.1 – цветовая палитра сайта.

На данном этапе были разработаны макеты страниц. Макеты страниц смотреть в приложении А.

## 2.3 Выбор шрифтового оформления

В данном проекте для в качестве основного шрифта использовался шрифт Mulish. Выбор в сторону этого шрифта был сделан в качестве придания сайту более эстетичного вида. Шрифт подключается с помощью правила @font-face и @import. Для мелких вставок некоторых элементов текста использованы следующие шрифты: Times New Roman, Georgia, Courier New.

Times New Roman - шрифт с засечками, разработанный для хорошей читабельности в плотных газетных столбцах. Он довольно сжатый, имеет большую высоту по оси Х и короткие нижние выносные элементы. Все это позволяет легко читать даже при небольшом размере и в плотных столбцах текста.

## 2.4 Разработка логотипа

Разработка логотипа осуществлялась в графическом редакторе Adobe Illustrator. Дизайн довольно минималистичен; было принято решение не загромождать его множеством декоративных элементов. Результат логотипа представлен на рисунке 2.2

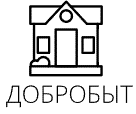


Рисунок 2.2 – логотип интернет-магазина.

## 2.5 Разработка пользовательских элементов

В проекте будут представлены такие элементы пользовательского интерфейса, как навигационное меню, кнопки, поле ввода информации пользователя, футер с ссылками на социальные сети, корзинка товаров.

На главной странице выступает навигационное меню, находящееся в хедере, меню представлено на рисунке 2.3.

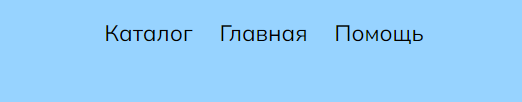


Рисунок 2.3 – Навигационное меню

Навигация – основной элемент для удобства сайта, это так называемая «карта», позволяющая перемещаться между различными страницами ресурса.

При нажатии на один из элементов навигационного меню происходит переход к определённому разделу сайта; при наведении на название категорий

Рассмотрим пользовательские элементы для оформления помощи со всплывающим модальным окном с формой, которую необходимо будет заполнить. На рисунке 2.4 представлена кнопка, при нажатии на которую пользователь увидит перед собой форму для оформления помощи на сайте.

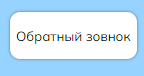


Рисунок 2.4 – Кнопка обратной связи

Форма обратной связи состоит из полей для ввода информации и кнопки для оформления. На рисунке 2.5, представлено поле ввода информации.

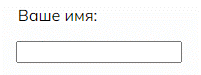


Рисунок 2.5 – Поле ввода информации для обратной связи

## 2.6 Разработка спецэффектов

На сайте разработаны динамические эффекты и анимации, которые придадут интерактивности сайту, что сделает его более привлекательным для пользователей.

Таким образом, при наведении на один из элементов в меню навигации, появляется подчеркивание с плавным переходом. Такое меню навигации представлено на рисунке 2.6.

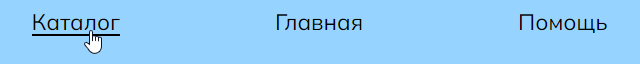


Рисунок 2.6 – Навигационное меню с эффектом

При наведении на одну из карточек товаров появляется тень позади самой карточки, картинки и текст увеличиваются в размере, чтобы вызвать у пользователя желание все рассмотреть. Так же при наведении на имя товара он меняет цвет.

В разделе «Каталог» и на главной странице в карточке товара присутствует кнопка добавления товара в корзину, при нажатии на которую товар и количество товаров появляется в корзине. В самой корзине представлена кнопка закрытия корзины, оформления заказа, а также очистки корзины. Пример оформления корзины с добавленными товарами представлен на рисунке 2.7.

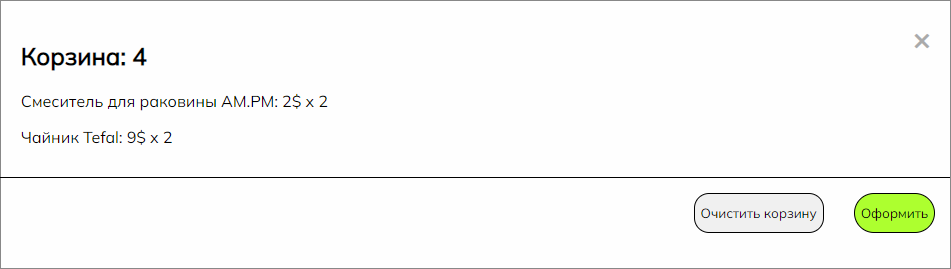


Рисунок 2.7 – Окно корзины

В разделе «Помощь» при наведении на один из вопросов текст будет плавно увеличиваться для придания плавности.

На второстепенных страницах присутствует все вышеперечисленное.

На мобильной версии главного сайта присутствуют все спецэффекты, что и на главной странице компьютерной версии, описанные ранее.

На мобильной версии второстепенных страниц присутствует все, что содержат второстепенные страницы компьютерной версии.

## 2.7 Выводы

В ходе данной главы была составлен прототипы структуры веб-сайта, показанный в приложении А, и разработаны макеты страниц сайта в Figma, показанные в приложении Б. Разобрав и проанализировав все выбранные аналоги, разработав логотип, пользовательские элементы и спецэффекты, а также подобрав нужную цветовую палитру вместе со шрифтами, был создан макет веб-сайта.

# 3. Реализация структуры веб-сайта

## 3.1. Структура HTML-документа

Разработка сайта – это долгая и плодотворная работа, которая, как правило, разделена на несколько этапов. Верстка сайта – это преобразование готового макета в продукт, состоящий из языка разметки HTML и CSS.

Следующий этап реализации проекта – верстка, в которой можно выделить несколько этапов:

* работа с макетом и сохранение изображений в отдельную папку;
* подключение шрифтов;
* разработка файлов с расширением. scss, компиляция в CSS;
* разработка страниц: HTML-верстка с подключением CSS;
* подключение скриптов, анимация элементов;
* разработка адаптивной верстки.

Графический контент был сохранен в подпапку каталога курсового проекта «Images», Компоненты интерфейса сайта сохранялись в корне данной подпапки, фотографии товаров и тому подобное. При сохранении фотографий использовался jpg-формат, обладающий насыщенными, яркими цветами, идеально подходящий для фотографий, требующих тщательной проработки цветовых деталей. Основные HTML-документы были сохранены в корневой папке «public», вся основная динамика, а также XML-документы были сохранены в папке «src». CSS-документы были сохранены в папку «Styles».

Структура HTML-документа для страницы: <!DOCTYPE html>, <html>. В теге <head> идёт подключение необходимых внешних файлов, указывается название страницы и некоторые метаданные. Формат изображен на рисунке 3.1. В теге <body> – контейнер, в котором содержится основная часть: она «разграничена» семантическими тегами - <header>, <footer> , <main> , <section>.

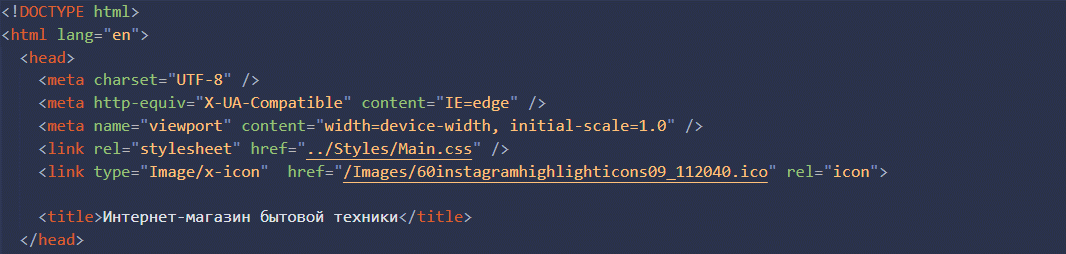


Рисунок 3.1 – Тег <head>

Создание верхней части веб-страницы начинается с создания элемента <header>, который является контейнером для всей информации. Затем создается контейнер, который задает максимальную ширину контейнера для удобного размещения содержимого на странице.

Секция заголовка содержит логотип и иконку корзины. Логотип представлен в виде изображения <img>. Иконка корзины представлена в виде SVG-изображения <svg>, которое задает форму и цвет иконки.

Секция заголовка содержит сам заголовок страницы "Интернет-магазин бытовой техники" в виде элемента <h1>.

Навигационное меню задается с помощью элемента <nav>, содержащего ссылки на другие страницы сайта в виде элементов <a>.

Формат содержимого тега <header> изображен на рисунке 3.2.

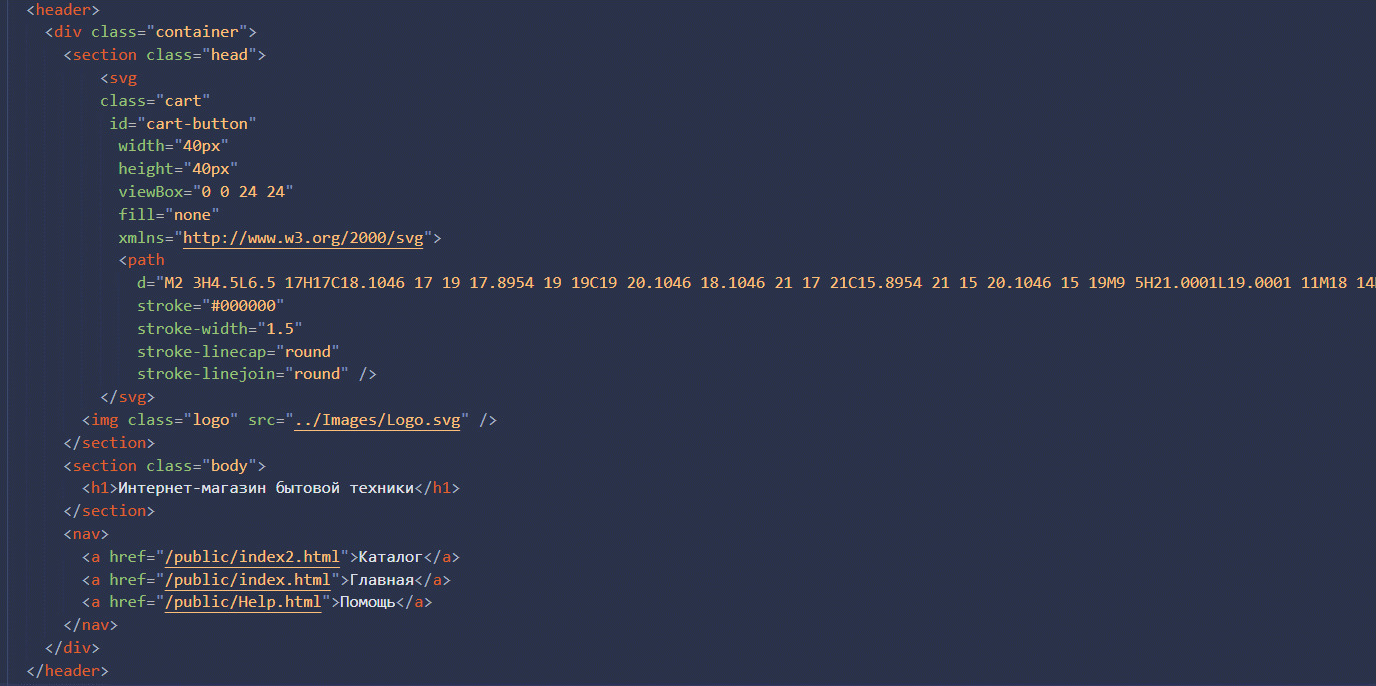


Рисунок 3.2 – Тег <header>

Верхняя часть веб-страницы дополнена элементами дизайна, такими как фон, цвета, шрифты и т.д., чтобы создать уникальный дизайн для сайта. Для этого была составлена таблица стилей показанная в приложении Г.

В проекте использовался самый актуальный вид верстки сайтов – блочный. Он основан на теге <div> и семантических тегах, с помощью которого создаются контейнеры, включающие в себя весь контент страницы или отдельного блока. Вся страница была мысленно разбита на смысловые секции, каждая из которых несет законченную мысль. В соответствии с выделенными блоками была создана основная структура разметки. Сайт можно разделить на несколько блоков: первый экран, навигация, обратная связь, корзина, список товаров, контакты – для каждого блока будет отведен свой тег <div> или соответствующий семантический тег, код главной страницы, на основе которой сделаны остальные, показан в приложении В и приложении Г.

Для тега <body> устанавливается значение «flex» к свойству «display», сделано это с целью создания свободных полей сайта.

Для тега <body> и всех входящих в него семантических тегов устанавливается значение «flex» к свойству «display». Это позволяет гибко управлять выравниванием, порядком, направлением и размерами элементов, особенно в контексте адаптивной вёрстки, когда размеры экранов могут существенно меняться.

Внутри тега <body> и <header> находятся два связанных элемента: кнопка добавления товара в корзину и SVG-картинка этой корзины. Для интерактивности взаимодействия и динамического изменения структуры страницы в секции со статьями, был реализован скрипт, приведенный в приложении Ж.

Данный скрипт дает пользователю возможность добавления товара в корзину. Каждый раз обновляя контейнер модального окна, который выступает в качестве корзины в HTML-документе.

Формат модального окна корзины изображен на листинге 3.3.

|  |
| --- |
| <div id="cart-modal" class="modal">  <div id="cart" class="modal-content">  <span class="close">&times;</span>  <h2 id="namec">Корзина</h2>  <span id="cart-list">Корзина пуста</span>  <div class="cart-foot">  <button id="clear-button">Очистить корзину</button>  <button class="zakaz">Оформить</button>  </div>  </div>  </div> |

Листинг 3.3 – Модальное окно корзины

Обновляя тег <span> при каждом клике на соответствующую кнопку товар добавляет в корзину как новый элемент страницы с тегом <p>.

На странице было так же использовано модальное окно для оформления заявки о помощи на сайте. Формат модального окна подобен предыдущем и представлен на листинге 3.4.

|  |
| --- |
| <div id="myModal" class="modal">  <div class="modal-content-form">  <span class="modal-close">&times;</span>  <h1>Оформить помощь</h1>  <form class="mod-form">  <legend>Ваше имя:</legend>  <input type="text">  <legend>Номер телефона: </legend>  <input type="text">  <legend>Почта: </legend>  <input type="email">  <legend>Опишите проблему: </legend>  <input type="text" class="desk-prob">  </form>  <div class="mod-footer">  <button class="help-btn">Запрос</button>  </div>  </div>  </div> |

Листинг 3.4 – Модальное окно обратной связи

Работа модальных окон в обоих случаях одинакова. Она заключается в изменении свойства «display» с «block» в «none» при клике на соответствующие кнопки и наоборот. Свойство также изменяется при клике пользователем в любое пустое место экрана. Подробнее о вышеперечисленном можно узнать в приложении В, приложении Г и приложении Ж.

## 3.2. Добавление таблиц стилей SCSS и CSS

Почти все элементы на сайте должны иметь визуальное оформление. Для визуального оформления использован CSS/SCSS. Они отвечают за то, как HTML-элементы будут выведены пользователю в браузере.

Существует несколько способов встраивания и подключения стилей: внешнее, внутреннее, строковое. В данном проекте используется внешнее подключение стилей с помощью тега link. Стили хранятся в отдельном файле, который может быть использован для любых веб-страниц. Данный способ один из самых удобных в реальной разработке. Так как стилей в проекте обычно много, то держать их в HTML-файле не очень удобно. Связано это с тем, что HTML-файл становится очень большим и ориентироваться в нем становится невозможно. Чтобы этого избежать, можно создать отдельный CSS-файл.  В таблицах стилей используются селекторы по тегу, по классу, по идентификатору.

Верстка была выполнена с использованием CSS Flexbox. Примеры использования CSS Flexbox на рисунке 3.5 для тега <header> и на рисунке 3.6 для тега <footer>.

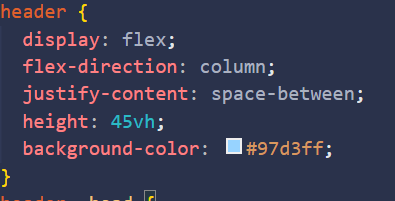


Рисунок 3.5 – Таблица стилей для тега <header>

Стили SCSS и CSS находятся во внешних файлах, и подключаются с помощью тега <link>. Всем страницам соответствуют файлы стилей SCSS.



Рисунок 3.6 - Таблица стилей для тега <footer>

В проекте используются классы, медиазапросы, псевдоклассы hover и active. Псевдокласс hover активизируется, когда курсор мыши находится в пределах элемента, но щелчка по нему не происходит. Псевдокласс active определяет стиль для активной ссылки. Активной ссылка становится при нажатии на нее.

Для элементов блочной модели и семантических тегов также была применена flex-вёрстка. Например формат контейнера <main> с основной информацией представлен на рисунке 3.7.

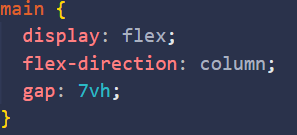


Рисунок 3.7 – Тег <main>

Такой же принцип был применен для таких блоков страницы как: навигационное меню, карточки товаров, контактная информация.

Для более гибкой разработки стилей моего сайта использовался препроцессор SCSS, который позволяет разработчикам использовать дополнительные функции, которые не поддерживаются в стандартном CSS. SCSS позволяет использовать переменные, вложенные правила, операторы, миксины, наследование и многое другое, что значительно упрощает процесс написания и поддержки кода CSS.

Одна из основных причин использования SCSS - это улучшение модульности и читаемости кода CSS. SCSS позволяет объединить несколько CSS файлов в один и упорядочить их, а также использовать переменные для быстрого и простого изменения стилей. SCSS также позволяет использовать вложенные правила, что значительно упрощает написание и поддержку кода, а также улучшает его читаемость. Пример использования вложенности представлен на листинге 3.8.

|  |
| --- |
| & .body {  display: flex;  justify-content: center;  & h1 {  margin: 0;  padding: 0;  font-size: 3.5em;  text-align: center;  }  } |

Листинг 3.8 – Вложенность для тега <body>

Для сайта была также применена адаптивная вёрстка.

Важность адаптивной верстки заключается в том, что в настоящее время большинство пользователей Интернета используют мобильные устройства для доступа к Интернету.

Поэтому, если веб-страница не будет адаптироваться к мобильным устройствам, это может привести к плохому пользовательскому опыту и снижению конверсии. Если же веб-страница будет адаптироваться к различным устройствам, это может улучшить пользовательский опыт и повысить конверсию.

Кроме того, адаптивная верстка позволяет создавать более доступный контент для пользователей с ограниченными возможностями, такими как слабовидящие или люди с ограниченными возможностями моторики. Также адаптивная верстка улучшает SEO (Search Engine Optimization) и повышает вероятность того, что веб-страница будет показана на первых страницах результатов поисковых систем.

В целом, адаптивная верстка является важным аспектом создания веб-страницы, который помогает повысить пользовательский опыт, улучшить SEO и увеличить конверсию.

Происходит это динамически, с использованием медиа-запросов, позволяющих автоматически определять разрешение монитора, и подставлять указанные значения. Медиа-запрос начинается с ключевого слова @media, где в качестве условия указывалась максимальная допустимая ширина.

Были прописаны медиа-запросы для экранов с максимальной шириной в 900 и минимальной 700 пикселей, а также с 700 и 400, 400 и 100 пикселей.

Для мобильного устройства больше использовалось максимально допустимое разрешение 700 и 400 пикселей. Пример формата представлен на листинге 3.8.

|  |
| --- |
| @media (max-width: 900px) and (min-width: 700px) {  header {  height: 40vh;  }  header .body h1 {  font-size: 3em;  }  header nav {  width: 45vw;  }  main .popular\_slider {  height: 50vh;  }  main .popular\_slider section {  height: 35vw;  }  main .popular\_slider span {  font-size: 2vw;  }  main .cards .card {  height: 25vh;  }  } |

Листинг 3.8 – Медиа-запрос для страницы каталога товаров

Прописывание медиа-запросов помогает в изменении положения объектов на странице. Пример адаптивной вёрстки представлен на рисунке 3.9.

Единственная проблема адаптивной вёрстки заключается в отсутствии возможности сжатия формата некоторых изображений до мобильного.

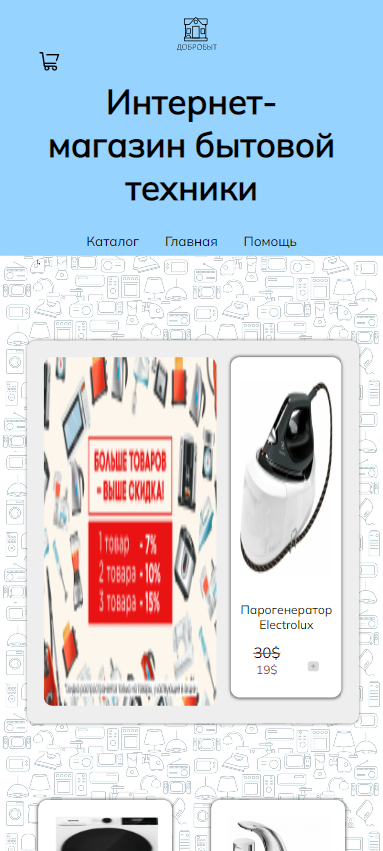


Рисунок 3.9 – Мобильная версия категории «Главная»

## 3.3. Использование стандартов XML (SVG)

Для хранения контактной информации из тега <footer> и информации о характеристиках каждого товара используются XML-документы. Для загрузки данных их XML-файлов используется код описанный в приложении Д.

Этот код получает параметры URL-адреса страницы и использует их для установки значения элементов на странице, а затем загружает XML-файл каталога, ищет элемент, соответствующий имени товара, полученному из URL-параметра, и устанавливает его описание в элемент на странице с идентификатором «charact». Если элемент не найден, ничего не происходит. Если при загрузке XML-файла произошла ошибка, выводится сообщение об ошибке в консоль.

Что касается контактной информации в теге <footer>, то этот код загружает XML-документ по указанному пути, затем с помощью объекта DOMParser парсит его содержимое в XML-дерево. Далее извлекает значения некоторых тегов из дерева с помощью метода getElementsByTagName, и заменяет содержимое элементов на странице с указанными идентификаторами на соответствующие значения. После этого происходит аналогичный процесс для других тегов, значения которых также извлекаются из XML-дерева и заменяют содержимое элементов на странице с указанными идентификаторами.

В целом расширяемый язык разметки позволяет определять и хранить данные совместно используемым способом. XML поддерживает обмен информацией между компьютерными системами, такими как веб-сайты, базы данных и сторонние приложения. Предопределенные правила упрощают передачу данных в виде XML-файлов по любой сети, поскольку получатель может использовать эти правила для точного и эффективного чтения данных. В проекте одна из страниц содержит XML-формат для хранения информации, используемой на странице.

SVG – это двухмерная графика и тем не менее это текстовый формат, который можно легко править в блокноте или просто рисовать в векторных редакторах. SVG одинаково хорошо выглядит без потери качества и на мобильном устройстве, и на стационарном мониторе домашнего ПК. Поэтому для создания логотипа, кнопки добавления в корзину, самой корзины был выбран формат SVG.

## 3.4 Выводы

В ходе данного раздела была разработана верстка сайта и стилевые решения сайта. Верстака сайта выполнялась с учетом необходимости наличия адаптивности, была разработана структура xml файлов, код для загрузки данных из xml файлов. Был разработан код для анимаций и основного функционала сайта.

# 4. Тестирование веб-сайта

## 4.1. Адаптивный дизайн веб-сайта

Адаптивность — одно из ключевых требований, предъявляемых к современным сайтам. Ресурс должен одинаково хорошо демонстрироваться на экране компьютера, планшете, смартфоне. Для этого создается адаптивный дизайн и используется адаптивная верстка. В итоге процесс веб-разработки усложняется, требует дополнительных ресурсов. Но результат стоит вложений: сайт отлично позиционируется на всех основных типах устройств, что гарантирует полноценный охват аудитории.

Для адаптивности использована flex-верстка для элементов страницы, а также ширина блоков задается максимально точно относительно других элементов, что позволяет им подстраиваться под ширину экрана пользователя. Flex-верстка используется для шапки сайта. Использованы медиа-запросы для изменения внешнего вида шапки для размещения блоков в необходимом порядке. Вид адаптивной шапки сайта изображен на рисунке 4.1.



Рисунок 4.1 – Вид шапки сайта в мобильной версии

Flex-верстка также используется для карточек товаров. Элементы адаптируются под ширину экрана пользователя. Пример масштабируемых карточек представлен на рисунке 4.2.

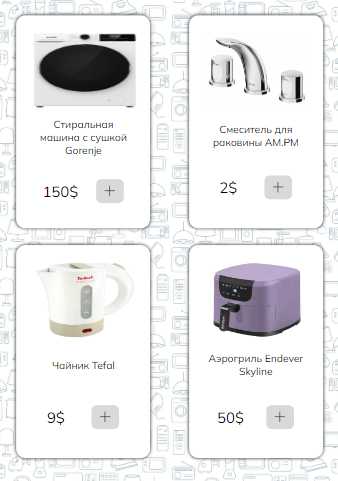


Рисунок 4.2 – Товары сайта в мобильной версии

Остальные элементы также хорошо адаптированы. Например на рисунке 4.3 изображена карточка с описанием товара.



Рисунок 4.3 – Характеристика товара в мобильной версии

В итоге сайт отлично позиционируется на всех основных типах устройств, что гарантирует полноценный охват аудитории.

## 4.2. Кроссбраузерность веб-сайта

Кроссбраузерность – новомодный термин в практике раскрутки и развития контентных сайтов, который на самом деле новым и не является. Его суть проста: кроссбраузерный ресурс одинаково удобен к прочтению зрителем на любой из существующих браузеров.

Кроссбраузерность предоставляет: улучшение поведенческих факторов (меньше отказов), увеличение продаж, формирование положительного имиджа бренда, расширение целевой аудитории.

Веб-сайт был открыт при помощи Google Chrome на рисунке 4.4.

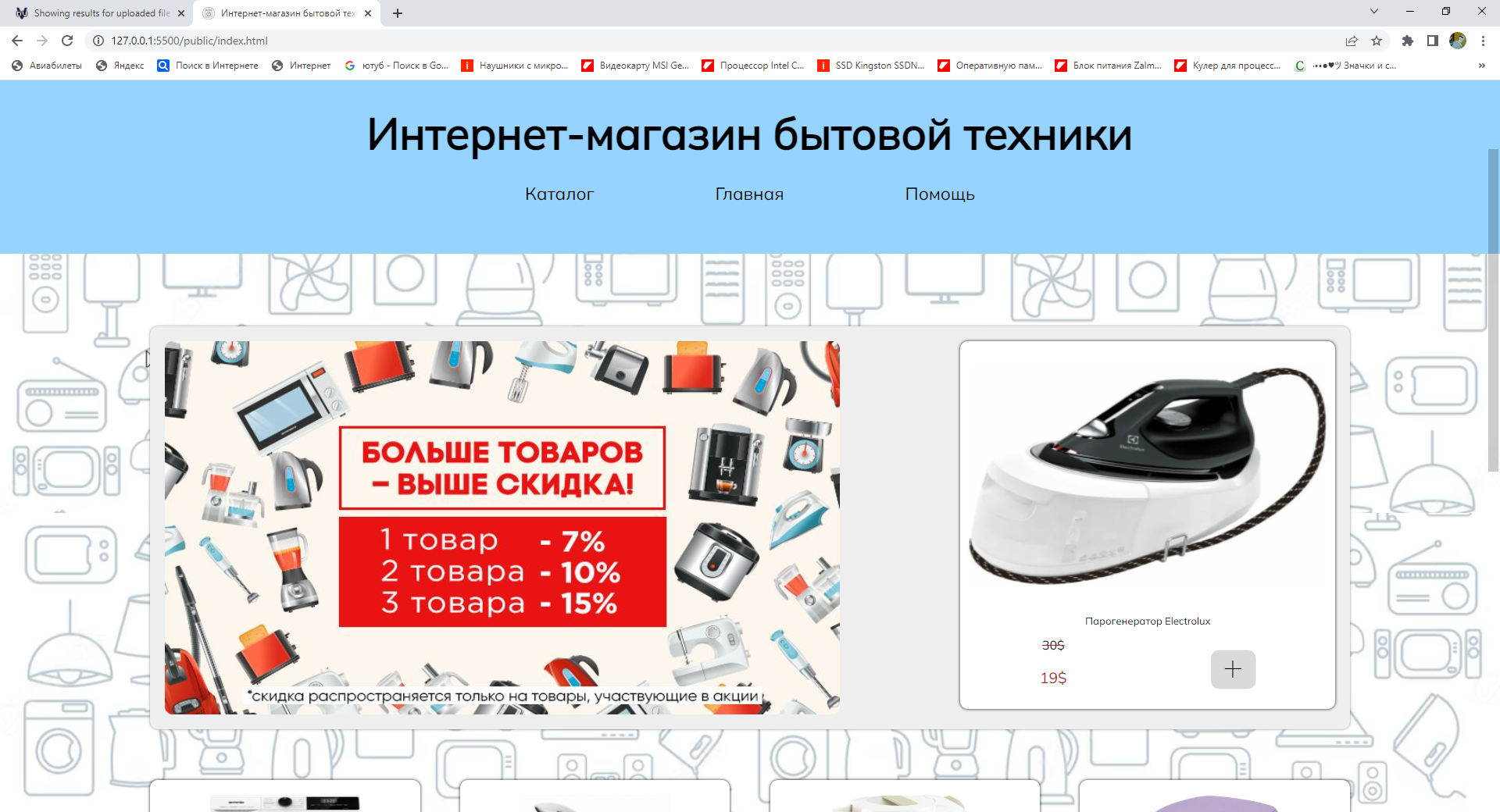


Рисунок 4.4 - веб-сайт, запущенный в Google Chrome

Проверку на кроссбраузерность на данном этапе разработки можно произвести только вручную, поэтому не исключается вероятность, что определенные свойства могут отображаться некорректно.

Сайт также был открыт в Microsoft Edge на рисунке 4.5.

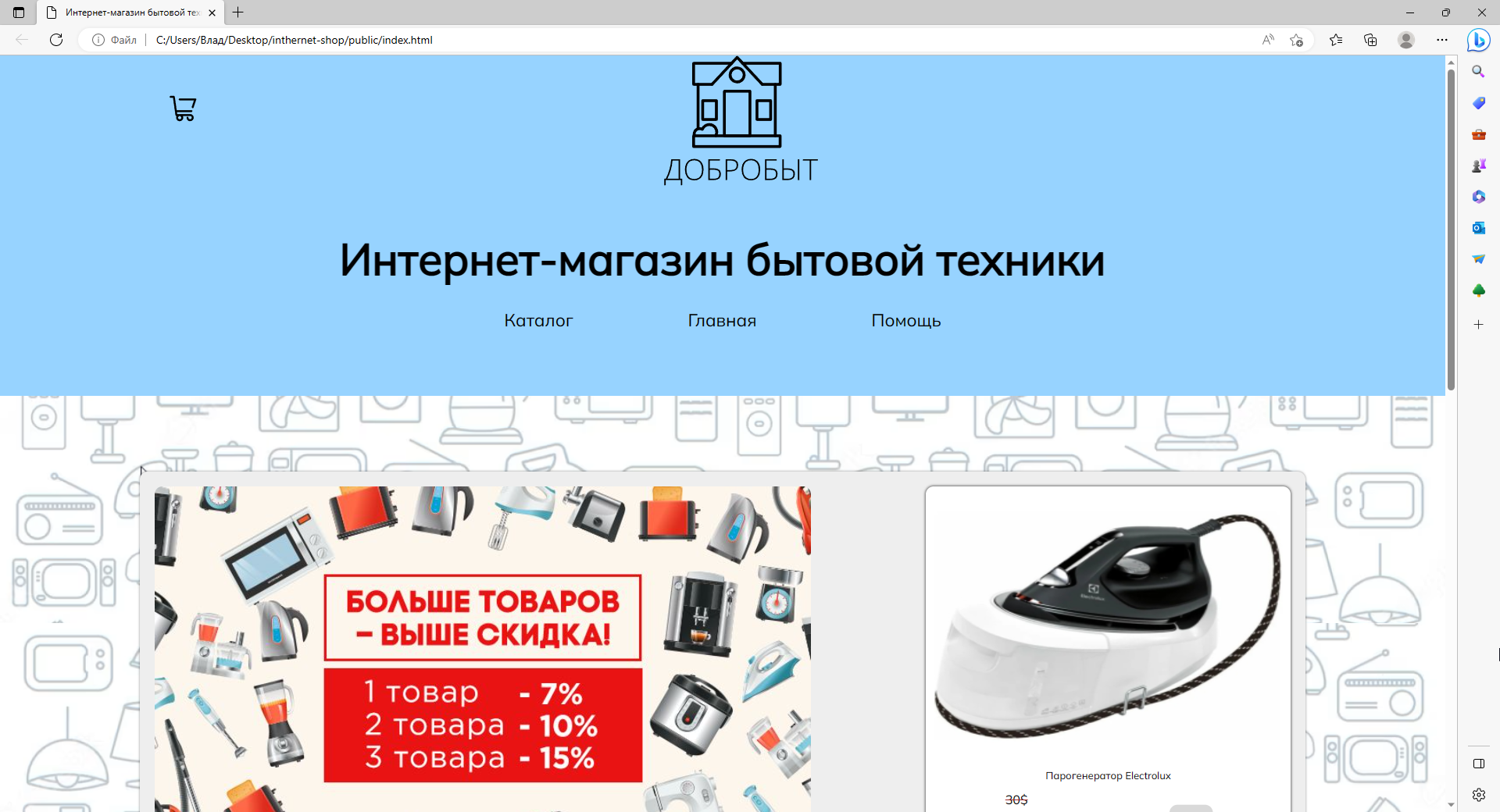


Рисунок 4.5 - веб-сайт, запущенный в Microsoft Edge

Сайт не обязательно верстать под все существующие браузеры и делать его отображение строго одинаковым во всех. Достаточно убедиться в нескольких моментах:

- информация читабельна и выводится правильно, элементы не перекрывают друг друга и не съезжают;

- отображается вся необходимая информация;

- сохранена структура сайта;

- отсутствуют недостатки вёрстки.

Сделать сайт кроссбраузерным — значит добиться схожей (не обязательно до мелочей) визуализации контента в популярных браузерах. Достичь такого результата можно несколькими путями.

## 4.3. Руководство пользователя

Для того чтобы перейти на другую страницу сайту откройте главное меню, для этого нужно перейти в начало страницы и найти навигационное меню. В нем есть определенная категория для главной страницы. Нажав на нее вы переместитесь на главную страницу. Верхушка сайта с навигационным меню представлено на рисунке 4.6.



Рисунок 4.6 – Верхушка сайта

Для просмотра товаров вам необходимо перейти в категорию «Каталог» в навигационном меню любой из страниц. Перейдя вы увидите список товаров, категории товаров, поисковое поле и сами карточки с товарами. Изображение каталога представлено на рисунке 4.7.

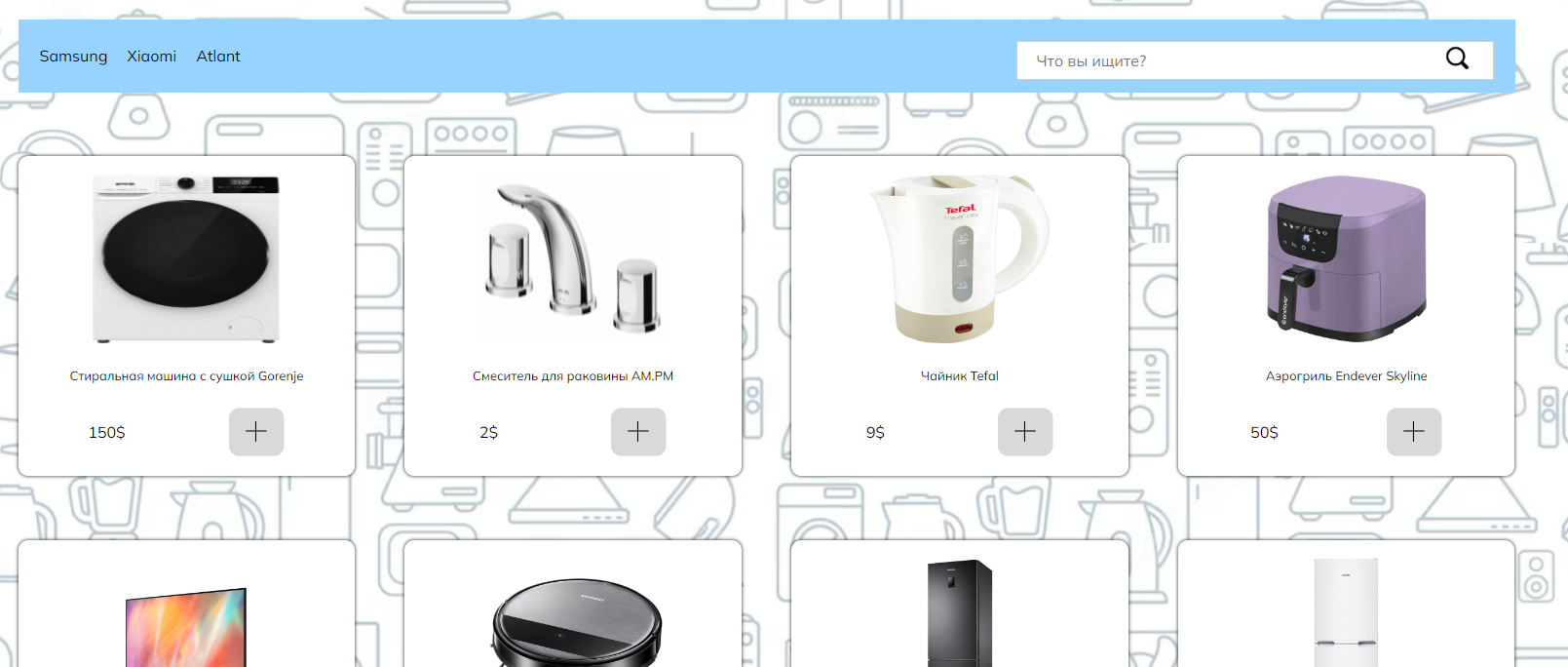


Рисунок 4.7 – Каталог товаров

Для того чтобы добавить товар в корзину нужно нажать соответствующую кнопку рядом с ценой.

Если нажать на имя товара то вы перейдете на его описание с характеристиками выбранного товара.

Перейдя во вкладку «Помощь» из навигационного меню, пользователь увидит ответы на часто задаваемые вопросы. Формат страницы изображен на рисунке 4.8.

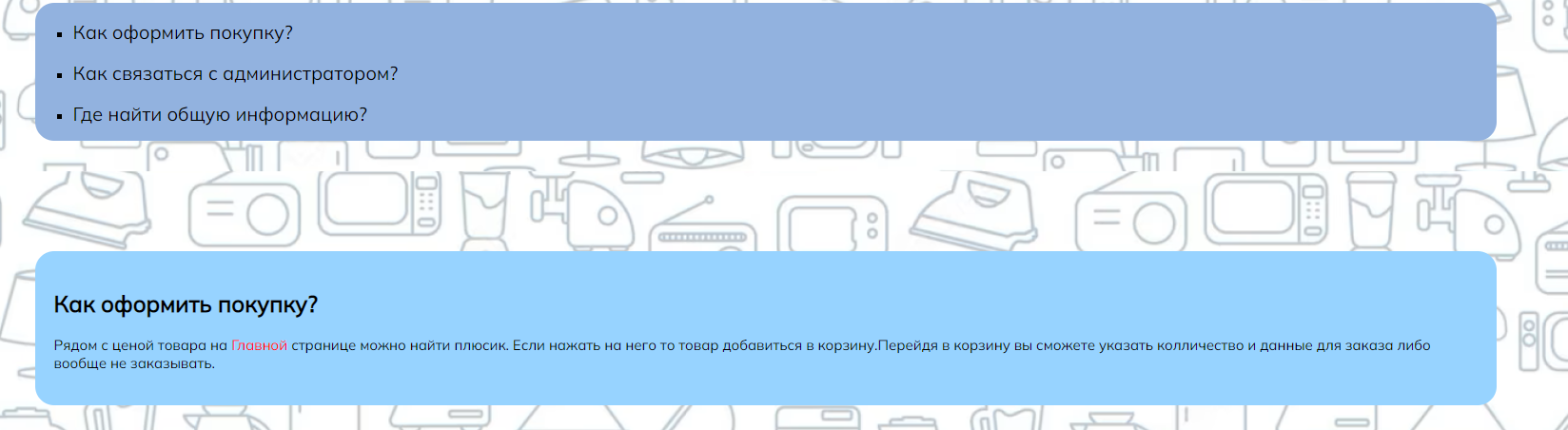


Рисунок 4.8 – Часто задаваемые вопросы

Как только пользователь добавил товар в корзину у него есть возможность оформить заказ выбранных товаров нажав на соответствующую кнопку в модульном окне корзины. Он также может удалить все выбранные товары.

## 4.4. Валидность

Валидность HTML-верстки – это её соответствие стандартам организации W3C. Отсутствие ошибок в вёрстке документа – один из основных показателей её качества. Автоматическая проверка вёрстки на ошибки была проведена в онлайн-сервисе W3C. Сайт прошёл проверку на валидность в сервисе validator.w3.org. Ошибок не обнаружено. Результат представлен на рисунке 4.9.

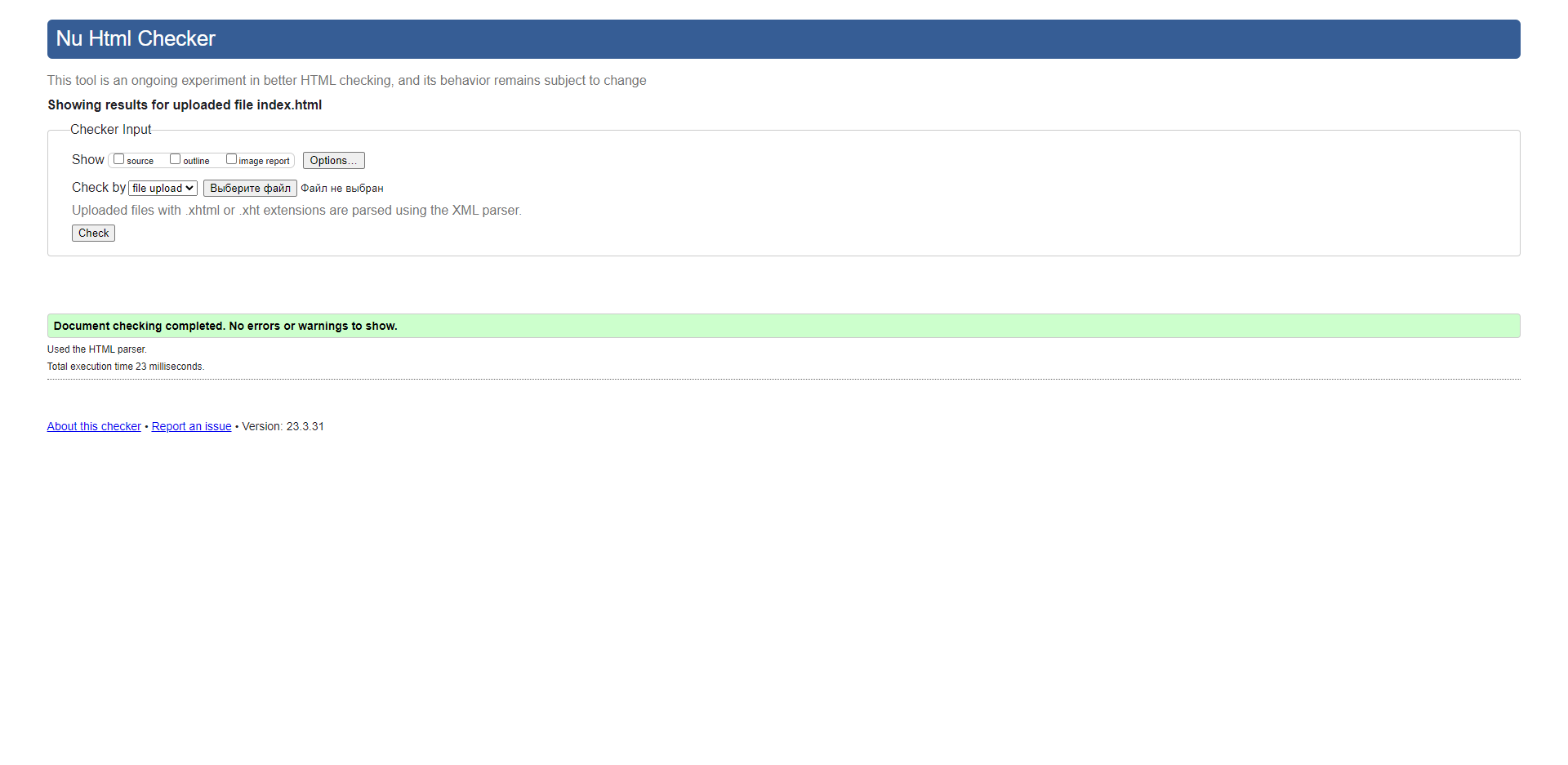


Рисунок 4.9 – Валидность сайта

## 4.4 Выводы

В данном разделе было проведено тестирование веб-сайта на адаптивность, кроссбраузерность, валидность. Также было написано руководство пользователя. В ходе тестирование стало понятно, что сайт без проблем работает на двух платформах, полностью адаптивен на мобильном устройстве. Проверка на валидность дала положительный результат.

# Заключение

В ходе курсового проекта был разработан веб-сайт «Интернет-магазин бытовой техники».

В процессе создания курсового проекта были изучены такие технологии как CSS Flexbox, улучшение навыков в работе с JavaScript, HTMK и СSS/SCSS. В ходе выполнения проекта был использован язык SCSS. Стили сайта прописаны в SCSS-файлах, которые компилируются в CSS с помощью препроцессора SASS. Технология Flexbox упрощает расположение элементов на странице и способствует созданию адаптивности верстки.

Были приобретены практические навыки в разработке программного проекта, кроссбраузерной адаптивной верстки для корректного отображения на устройствах с различной шириной экрана. Также был приобретен опыт в составлении технических задач и развита способность продумывать действия пользователя, его мышление, чтобы в последствии реализовать максимально релевантный продукт. Были улучшены навыки работы с редактором Visual Studio Code и его функциями.

Был проведен анализ веб-сайтов данной тематики для выявления достоинств и недостатков, которые помогли определиться с методами и способами для реализации проекта.

Работа с макетом производилась в сервисе Figma.

Для выполнения поставленной цели курсового проекта были реализованы следующие задачи:

1. Анализ аналогов.
2. Анализ существующих языков разметки, инструментов библиотеки.
3. Разработка макета и прототипа.
4. Выбор способа верстки.
5. Выбор стилевого оформления.
6. Разработка логотипа.
7. Разработка основной структуры сайта.
8. Разработка пользовательских элементов.
9. Разработка спецэффектов.
10. Добавление веб-сайту адаптивности.
11. Кроссбраузерность веб-сайта.
12. Тестирование веб-сайта.
13. Создание руководства пользователя.

По завершению работы над программным продуктом был создан кроссбраузерный адаптивный сайт, который был загружен на GitHub репозиторий, для возможности продемонстрировать сайт любому пользователю и любым пользователем, был получен опыт и новые знания в верстке веб-сайтов, изучены стандарты HTML5, выполнены цели и задачи.

# Список использованных литературных источников

* 1. 5 элемент - интернет-магазин электроники в Беларуси [Электронный ресурс] – Режим доступа: https://5element.by/ – Дата доступа: 13.04.2023.
  2. Электросила - интернет-магазин электроники в Беларуси [Электронный ресурс] – Режим доступа: https://sila.by/ – Дата доступа: 13.04.2023.
  3. 21vek - интернет-магазин в Беларуси [Электронный ресурс] – Режим доступа: https://21vek.by/ – Дата доступа: 13.04.2023.
  4. Html5book [Электронный ресурс] – Режим доступа: <https://html5book.ru/css3-flexbox/> – Дата доступа: 13.04.2023.
  5. Web-creator [Электронный ресурс] – Режим доступа: https://web-creator.ru/articles/html\_validity\_and\_semantically – Дата доступа: 13.04.2023.
  6. The W3C Markup Validation Service [Электронный ресурс] – Режим доступа: https://validator.w3.org/ – Дата доступа: 23.04.2023.
  7. Документация по Figma [Электронный ресурс]. — Режим доступа: <https://help.figma.com> - Дата доступа: 13.04.2023
  8. Документация по Sass/SCSS [Электронный ресурс]. — Режим доступа: <https://sass-scss.ru/guide/> - Дата доступа: 13.04.2023

# Приложение А Прототипы веб-страниц

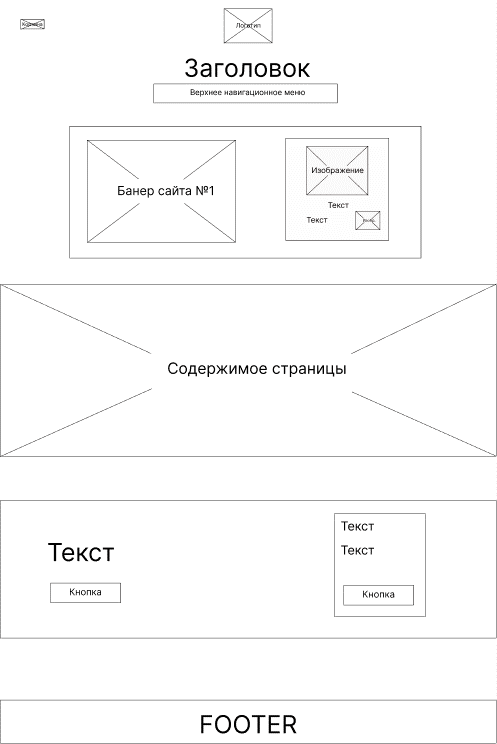


Рисунок 1 – Прототип главной страницы

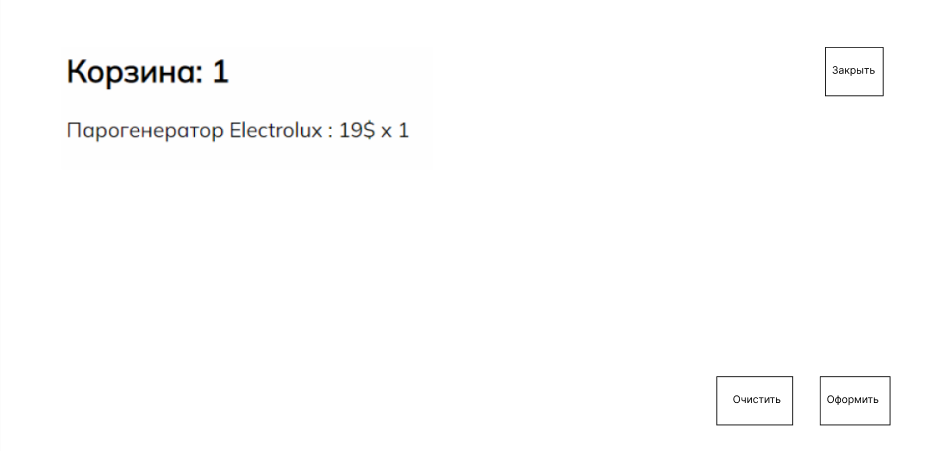


Рисунок 2 – Прототип страницы корзины

# Приложение Б Макет структуры веб-сайта

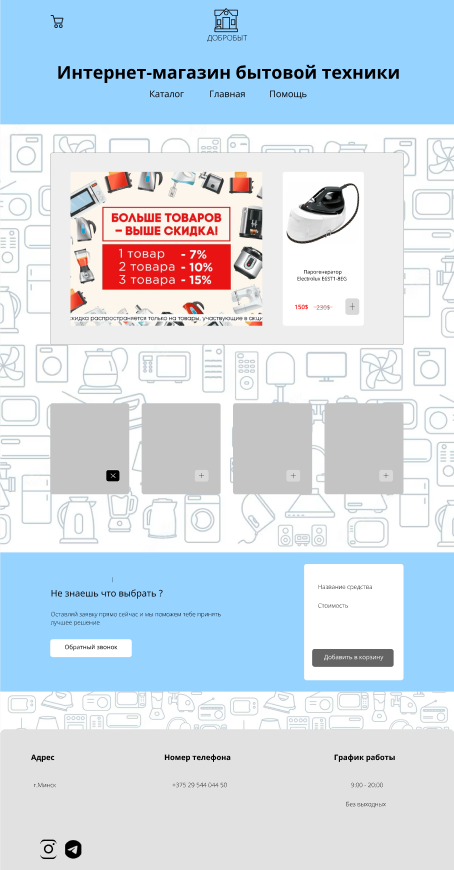


Рисунок 1 – Макет главной страницы

# Приложение В Листинг HTML-документа

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8" />  <meta http-equiv="X-UA-Compatible" content="IE=edge" />  <meta name="viewport" content="width=device-width, initial-scale=1.0" />  <link rel="stylesheet" href="../Styles/Main.css" />  <link type="Image/x-icon" href="/Images/60instagramhighlighticons09\_112040.ico" rel="icon">  <title>Интернет-магазин бытовой техники</title>  </head>  <body>  <header>  <div class="container">  <section class="head">  <svg  class="cart"  id="cart-button"  width="40px"  height="40px"  viewBox="0 0 24 24"  fill="none"  xmlns="http://www.w3.org/2000/svg">  </svg>  <img class="logo" src="../Images/Logo.svg" />  </section>  <section class="body">  <h1>Интернет-магазин бытовой техники</h1>  </section>  <nav>  <a href="/public/index2.html">Каталог</a>  <a href="/public/index.html">Главная</a>  <a href="/public/Help.html">Помощь</a>  </nav>  </div>  </header>  <div class="container">  <main>  <section class="popular\_slider">  <img class="for\_cell" src="../Images/18\_1678702144.jpg" />  <div class="card ">  <section class="slider-card">  <img class="item-photo" src="../Images/parogenerator.png" />  <p class="item-name">Парогенератор Electrolux  </p>  <div class="gets">  <div class="cost">  <span>30$</span>  <p class="price">  19$  </p>  </div>  <img class="button" src="../Images/Button.svg" />  </div>    </section>  </div>  </section>  <section class="cards">  <div class="card">  <img class="item-img" src="../Images/стиралка.jpg"/>  <p class="item-name">Стиральная машина с сушкой Gorenje</p>  <div class="item-get">  <p class="price">  150$ </p>  <img class="button" src="../Images/Button.svg" />  </div>  </div>  <div class="card">  <img class="item-img" src="../Images/кран.jpeg"/>  <p class="item-name">Смеситель для раковины AM.PM</p>  <div class="item-get">  <p class="price">  2$ </p>  <img class="button" src="../Images/Button.svg" />  </div>  </div>  <div class="card">  <img class="item-img" src="../Images/чайник.jpg"/>  <p class="item-name">Чайник Tefal</p>  <div class="item-get">  <p class="price">  9$ </p>  <img class="button" src="../Images/Button.svg" />  </div>  </div>  <div class="card">  <img class="item-img" src="../Images/аэрогриль.jpg"/>  <p class="item-name">Аэрогриль Endever Skyline</p>  <div class="item-get">  <p class="price">  50$ </p>  <img class="button" src="../Images/Button.svg" />  </div>  </div>  </section>  </main>  </div>  <section class="info">  <div class="container">  <section class="information">  <section class="text">  <h1>Не знаешь что выбрать ?</h1>  <p>  Оставляй заявку прямо сейчас и мы поможем тебе принять лучшее  решение  </p>  <button id="call">Обратный зовнок</button>  </section>  <section class="cart\_setter">  <input id="get-name" type="text" placeholder="Название товара">  <input id="get-cost" type="text" placeholder="Стоимость">  <button id="get-card">Добавить в очередь</button>  </section>  </section>  </div>  </section>  <footer>  <section>  <div>  <h3 id="adHead"></h3>  <p id="address"></p>  <div class="contacts">  <a href="https://www.instagram.com/vladik\_vodopadik25/">  <svg  class="intsa"  width="800px"  height="800px"  viewBox="0 0 24 24"  fill="none"  xmlns="http://www.w3.org/2000/svg">  </svg>  </a>  <a href="https://t.me/vladislav2025">  <svg  class="telegram"  width="800px"  height="800px"  viewBox="0 0 24 24"  fill="none"  xmlns="http://www.w3.org/2000/svg">  </svg>  </a>  </div>  </div>  <div>  <h3 id="phHead"></h3>  <p id="phone"></p>  </div>  <div>  <h3 id="wHead"></h3>  <p id="work-time"></p>  <p id="wkd"></p>  </div>  </section>  </footer>  <div id="cart-modal" class="modal">  <div id="cart" class="modal-content">  <span class="close">&times;</span>  <h2 id="namec">Корзина</h2>  <span id="cart-list">Корзина пуста</span>  <div class="cart-foot">  <button id="clear-button">Очистить корзину</button>  <button class="zakaz">Оформить</button>  </div>  </div>  </div>  <div id="myModal" class="modal">  <div class="modal-content-form">  <span class="modal-close">&times;</span>  <h1>Оформить помощь</h1>  <form class="mod-form">  <legend>Ваше имя:</legend>  <input type="text">  <legend>Номер телефона: </legend>  <input type="text">  <legend>Почта: </legend>  <input type="email">  <legend>Опишите проблему: </legend>  <input type="text" class="desk-prob">  </form>  <div class="mod-footer">  <button class="help-btn">Запрос</button>  </div>  </div>  </div>  <script src="/src/itemInfo.js">  </script>  <script src="/src/storage.js">  </script>  <script src="/src/index.js">  </script>  </body>  </html> |

Листинг 1 – файл index.html

# Приложение Г Листинг SCSS и CSS

|  |
| --- |
| @media (max-width: 900px) and (min-width: 700px) {  header {  height: 40vh;  }  header .body h1 {  font-size: 3em;  }  header nav {  width: 45vw;  }  main .popular\_slider {  height: 50vh;  }  main .popular\_slider section {  height: 35vw;  }  main .popular\_slider span {  font-size: 2vw;  }  main .cards .card {  height: 25vh;  }  }  @media (max-width: 700px) and (min-width: 400px) {  header {  height: 30vh;  }  header .body h1 {  font-size: 2.3em;  }  header nav {  width: 55vw;  }  header nav a {  font-size: 0.9em;  }  main .popular\_slider section .name {  margin-top: 0;  font-size: 0.2em;  }  main .popular\_slider section .price {  margin-top: 0;  font-size: 0.8em;  }  main .popular\_slider {  height: 45vh;  width: 39vh;  margin-left: -3vw;  }  main .popular\_slider .for\_cell {  width: 45vw;  }  main .popular\_slider section {  margin-left: 10px;  height: 40vh;  width: 13vh;  }  #get-card {  height: 35px;  }  #get-name {  width: 75px;  }  main .cards {  flex-wrap: wrap;  }  main .cards .card {  margin-top: 2vh;  width: 35vw;  height: 25vh;  }  main .cards .card .item-img {  width: 150px;  height: 80px;  }  main .cards .card .item-name {  font-size: 11px;  }  main .cards .card .item-img {  width: 110px;  }  main .cards .card .button {  width: 30px;  }  footer section div h3 {  font-size: 0.9em;  }  footer section div p {  font-size: 0.9em;  }  .categories p {  margin-left: 9px;  font-size: 5vw;  cursor: pointer;  }  input#search-bar {  display: none;  }  .modal-content-form {  width: 300px;  }  .item-info {  border: 1px solid black;  border-radius: 10px;  background-color: white;  height: 650px;  }  .item-info img {  width: 200px;  height: 220px;  }  .item-info .lead {  margin-left: 50px;  }  .item-info .lead .bot {  display: block;  }  footer section div .contacts svg {  height: 15vw;  width: 10vw;  }  }  @media (max-width: 400px) and (min-width: 100px) {  header {  height: 30vh;  }  header .head .logo {  width: 42vw;  }  header .head .cart {  width: 26vw;  }  header .body h1 {  font-size: 1.8em;  }  header nav {  width: 55vw;  }  header nav a {  font-size: 0.7em;  }  main {  gap: 3vh;  }  main .popular\_slider section .name {  margin-top: 0;  font-size: 0.6em;  }  main .popular\_slider section .price {  margin-top: 0;  font-size: 0.6em;  }  main .popular\_slider {  height: 30vh;  }  main .cards {  flex-wrap: wrap;  }  main .cards .card {  margin-top: 5vh;  width: 35vw;  height: 20vh;  }  footer section {  height: 35vh;  }  footer section div h3 {  font-size: 0.7em;  }  footer section div p {  font-size: 0.6em;  } |

Листинг 1 – файл Main.css

# Приложение Д Листинг кода для работы с XML-файлами

|  |
| --- |
| const urlParams = new URLSearchParams(window.location.search);  const itemName = urlParams.get('name');  const itemPrice = urlParams.get('price');  const itemImgSrc = urlParams.get('img');  const itemNameElement = document.querySelector('.item-name');  const itemPriceElement = document.querySelector('.item-price');  const itemImgElement = document.querySelector('.item-img');  itemNameElement.textContent = itemName;  itemPriceElement.textContent = itemPrice;  itemImgElement.src = itemImgSrc;  const ch = document.getElementById("charact");  fetch('/src/catalog.xml')  .then(response => response.text())  .then(data => {  const parser = new DOMParser();  const xmlDoc = parser.parseFromString(data, "text/xml");  const items = xmlDoc.getElementsByTagName("item");  for (let i = 0; i < items.length; i++) {  const name = items[i].getAttribute("name");  if (name === itemName) {  const description = items[i].getElementsByTagName("description")[0].innerHTML;  ch.innerHTML = description;  break;  }  }  })  .catch(error => console.error(error)); |

Листинг 1 – файле getParams.js

|  |
| --- |
| fetch('/src/data.xml')  .then(response => response.text())  .then(xml => {  const parser = new DOMParser();  const xmlDoc = parser.parseFromString(xml, "text/xml");  // part 1  const adHead = xmlDoc.getElementsByTagName("adHead")[0].childNodes[0].nodeValue;  const phHead = xmlDoc.getElementsByTagName("phHead")[0].childNodes[0].nodeValue;  const wHead = xmlDoc.getElementsByTagName("wHead")[0].childNodes[0].nodeValue;  const sumAd = document.getElementById("adHead");  sumAd.innerHTML = `${adHead}`  const sumPh = document.getElementById("phHead");  sumPh.innerHTML = `${phHead}`  const sumW = document.getElementById("wHead");  sumW.innerHTML = `${wHead}`;  // part 2  const address = xmlDoc.getElementsByTagName("address")[0].childNodes[0].nodeValue;  const phone = xmlDoc.getElementsByTagName("phone")[0].childNodes[0].nodeValue;  const time = xmlDoc.getElementsByTagName("time")[0].childNodes[0].nodeValue;  const wkd = xmlDoc.getElementsByTagName("weekend")[0].childNodes[0].nodeValue;    // Вставляем данные в footer  const addressEl = document.getElementById('address');  addressEl.innerHTML = `${address}`;  const phoneEl = document.getElementById('phone');  phoneEl.innerHTML = `${phone}`;  const timeEl = document.getElementById('work-time');  timeEl.innerHTML = `${time}`;  const wkdEl = document.getElementById('wkd');  wkdEl.innerHTML = `${wkd}`;  }); |

Листинг 2 – Файл storage,js

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <catalog>  <item name="Стиральная машина с сушкой Gorenje">  <description>  Вес загрузки белья: 7 кг  Глубина: 65.1 см  Класс энергопотребления: A++  Скорость отжима до: 800 об/мин  Количество программ стирки: 16  </description>  </item>  <item name="Смеситель для раковины AM.PM">  <description>  Тип водонагревателя: Проточный  Мощность: 3 кВт  </description>  </item>  <item name="Чайник Tefal">  <description>  Нагревательный элемент: Закрытый  Объем воды: 1.7 л  Материал корпуса: Пластик  </description>  </item>  <item name="Парогенератор Electrolux">  <description>  Максимальная мощность: 2000 Вт  Рабочая поверхность: Resilium, Анодированная  Емкость резервуара для воды: 1200 мл  Давление пара: 6  Паровой удар: Да  </description>  </item>  <item name="Аэрогриль Endever Skyline">  <description>  Мощность: 1500 Вт  Объем чаши: 2.5 л  </description>  </item>  <item name="Телевизор Samsung">  <description>  Тип экрана: LCD  Диагональ: 43  Разрешение: Ultra HD 4K-3840x2160 пикс.  Smart TV: Да  Операционная система: Tizen  </description>  </item>  <item name="Робот-пылесос SAMSUNG">  <description>  Тип уборки: Сухая, Влажная  Объем пылесборника: 0.2 л  Ориентация в пространстве: Гироскопические сенсоры, Инфракрасные сенсоры, Контактные датчики  Датчики: Определения препятствий, Определения ступенек, Определение обрыва  Время работы от аккумулятора (макс): 150 мин  </description>  </item>  <item name="Холодильник Samsung">  <description>  No Frost: Полный  Полезный объем: 367 л, 269 л + 98 л  Цвет: Черный  Тип управления: Электронное  Класс энергопотребления: A+  Высота: 201 см  </description>  </item>  <item name="Холодильник ATLANT">  <description>  Полезный объем: 198 л, 131 л + 67 л  Цвет: Белый  Тип управления: Электромеханическое  Класс энергопотребления: A  Высота: 161.5 см  </description>  </item>  </catalog> |

Листинг 3 – Файл catalog.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <contacts>  <adHead>Адрес:</adHead>  <phHead>Номер телефона:</phHead>  <wHead>Графиик работы:</wHead>  <contact>  <address>г.Минск</address>  <phone>+375 29 123-45-67</phone>  <time>9.00-20.00</time>  <weekend>Без выходных</weekend>  </contact>  </contacts> |

Листинг 4 – файл data.xml

# Приложение Ж Листинг JavaScript

|  |
| --- |
| const cart = {};  function addToCart(name, price, cart) { // корзина с параметрами-переменными  if (cart[name]) { // если товар уже есть в корзине  cart[name].count++; // увеличиваем его количество  } else { // если товара еще нет в корзине  const item = { name, price, count: 1 };  cart[name] = item; // добавляем товар в корзину  }  console.log(cart)  updateCart(cart); // вызваем функцию для получения данных  }  function updateCart(cart) { // новая функция которая обновляет корзину  const cartElement = document.getElementById("namec"); // доступ к области который представляет из себя корзину  const count = Object.values(cart).reduce((acc, item) => acc + item.count, 0); // считаем общее количество товаров  cartElement.innerHTML = `Корзина: ${count}`; // выводим в область (сама область внутри модульного окна)  const cartList = document.getElementById("cart-list"); // доступ к области модульного окна которое будет содержать в себе товары  cartList.innerHTML = ""; // пока сделаем его пустым  for (const name in cart) { // проходим по всем товарам в корзине  const item = cart[name]; // переменная для каждого предмета в корзине  const par = document.createElement("p"); // создаем параграф для елементов  par.textContent = `${item.name}: ${item.price} x ${item.count}`; // выводим название, цену и количество товара  cartList.appendChild(par); // каждый раз вставляем новый параграф как ребенка к родителю  }  }  const buttons = document.querySelectorAll(".button"); // получаем доступ к кнопке которая добавляет товар в корзину типо плюсик  buttons.forEach((button) => { // для каждой этой кнопки если их несколько  button.addEventListener("click", () => { // добавляем событие клика  const card = button.closest(".card"); // тут будут переменные для корзины , имени товара и цены  const name = card.querySelector(".item-name").textContent;  const price = card.querySelector(".price").textContent;  addToCart(name, price, cart); // вызываем фукнцию добавления при клике  });  });  // Открыть модальное окно при клике на кнопку "Корзина"  const cartButton = document.getElementById("cart-button");  const cartModal = document.getElementById("cart-modal");  const cartModalClose = document.getElementsByClassName("close")[0];  cartButton.addEventListener("click", () => {  cartModal.style.display = "block";  });  // Закрыть модальное окно при клике на кнопку "X"  cartModalClose.addEventListener("click", () => {  cartModal.style.display = "none";  });  // Закрыть модальное окно при клике вне его области  window.addEventListener("click", (event) => {  if (event.target == cartModal) {  cartModal.style.display = "none";  }  });  const clearButton = document.getElementById("clear-button");  clearButton.addEventListener("click", () => {  for (const name in cart) {  delete cart[name]; // удаляем все товары из корзины  }  updateCart(cart); // вызываем функцию обновления корзины  });  // Find item ==>  let cardget = document.getElementById("get-card");  cardget.onclick = function(){  let item\_get\_name = document.getElementById("get-name").value  let item\_get\_price = document.getElementById("get-cost").value    console.log(item\_get\_name +" "+ item\_get\_price + "$")    }  // modal #2  // Получаем ссылки на элементы страницы  var modal = document.getElementById("myModal");  var btn = document.getElementById("call");  var span = document.getElementsByClassName("modal-close")[0];  // Показываем модальное окно при нажатии на кнопку  btn.onclick = function() {  modal.style.display = "block";  }  // Скрываем модальное окно при нажатии на крестик  span.onclick = function() {  modal.style.display = "none";  }  // Скрываем модальное окно при нажатии за его пределами  window.onclick = function(event) {  if (event.target == modal) {  modal.style.display = "none";  }  } |

Листинг 1 – Файл index.js